

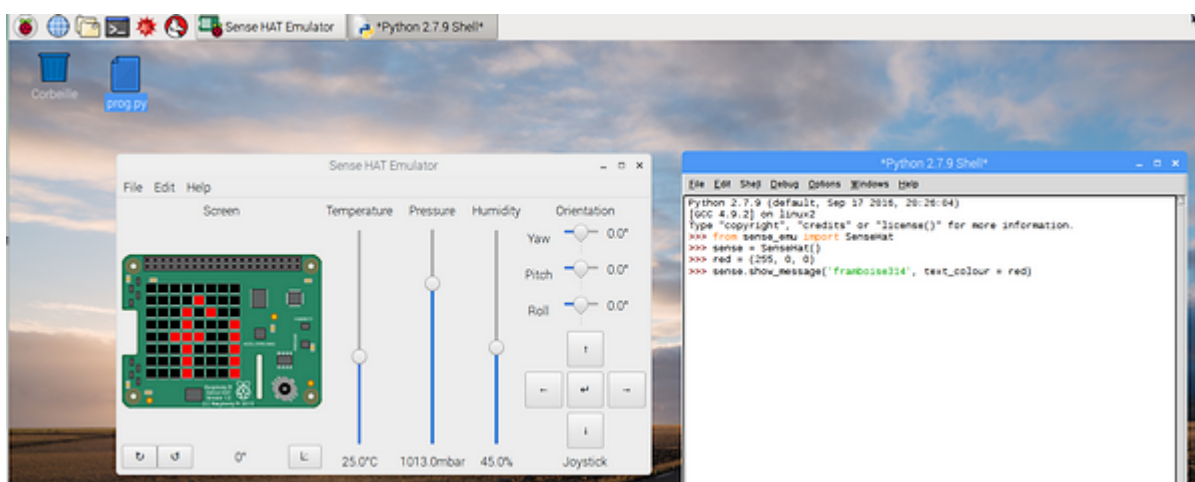
Ceci est la doc du Sense-hat

Le Sense-hat dans les étoiles : Astro-pi



Sense-hat

Une video



Un

émulateur de matrice LEDs Sense-Hat

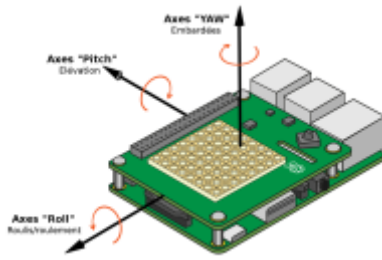
C'est quoi un Sense-Hat

Le Raspberry Pi Hat Sense est le produit idéal, grâce à ses nombreux capteurs, pour en apprendre davantage sur la programmation et la façon dont nous interagissons avec le monde qui nous entoure.

Le Hat Sense se fixe sur le dessus du Raspberry Pi via les 40 broches GPIO. Ses capteurs de circuits intégrés peuvent être utilisés pour de nombreux type d'expériences, applications, et même jeux.

Le Sense HAT supporte toute une série de projets pour le Pi de framboise, il peut mesurer la vitesse du Pi (Il mesure votre vitesse...), Quelle est la température ? Le taux d'humidité ? (Humidité de l'air), Quelle direction vous prenez ? Et bien plus.

Le 8x8 LED Matrix vous permet d'afficher les données des différents capteurs, il peut vous montrer quel est géomagnétique Nord en programmant une boussole à l'aide du magnétomètre, ou simplement être utilisé pour jouer à des jeux comme Tetris, Pong et Snake avec le joystick. Le joystick peut également être utilisé pour permettre à un utilisateur humain d'interagir avec les programmes exécutés sur le FAT de framboise Pi Sense-Hat



Détails techniques

Gyroscope

Capteur de vitesse angulaire (dps): ~ 245/500/2000

Accéléromètre

Capteur d'accélération linéaire (g): ~ 2/4/8/16

Magnétomètre

Capteur magnétique (gauss): ~ 4/8/12/16

Baromètre: 260 - 1260 hPa plage absolue (la précision dépend de la température et de la pression, ~ 0,1 hPa dans des conditions normales)

Capteur de température: précis à ~ 2 ° C dans la plage 0-65 ° C

Capteur d'humidité relative: précis à ~ 4,5% dans la gamme 20-80% rH, précis à ~ 0,5 ° C dans la gamme 15-40 ° C

Affichage matriciel 8x8 LED

Petit joystick à 5 boutons

Les séparateurs peuvent venir en plastique ou en métal

Dimensions du produit: 65.1mm x 56.6mm x 13.9mm / 2.6 "x 2.2" x 0.5 "

Poids du produit: 20.4g / 0.7oz

le schéma du Sense-Hat

[sense-hat-v1_0.pdf](#)

la programmation

[Debuter en Python](#)

[Des exemples sur github](#)

[documentation > hardware > sense-hat](#)

[Référence des API Sense Hat](#)

La librairie des exemples Hense-hat

Les programmes d'essai d'Henri

[texteBleu.py](#)

```
#!/usr/bin/python
from sense_hat import SenseHat

sense = SenseHat()
sense.set_rotation(180)
blue = (0, 0, 255)
sense.show_message("Un petit pas pour le R_Pi", text_colour=blue)
```

[environnement.py](#)

```
#Environnement
#!/usr/bin/python
from sense_hat import SenseHat
import time

sense = SenseHat()

while True:

    t = sense.get_temperature()
    p = sense.get_pressure()

    h = sense.get_humidity()

    t = round(t, 1) #arrondi à un chiffre après la virgule
    p = round(p, 1)
    h = round(h, 1)

    msg = "Temperature = %s, Pression=%s, Humidity= %s" % (t,p,h)
    sense.show_message (msg, scroll_speed=0.1)
```

[mouvement.py](#)

```
#motions
#!/usr/bin/python
from sense_hat import SenseHat
import time

sense = SenseHat()

while True:
    pitch, roll, yaw = sense.get_orientation(). values()
    print ("Elevation/pitch=%s, Roulis/roll=%s, Embardee/yaw=%s" %
    (pitch,yaw, roll))

    time.sleep(2)
```

[temperature.py](#)

```
#Temperature
#!/usr/bin/python
from sense_hat import SenseHat

sense = SenseHat()

temp = sense.get_temperature()
print("Temperature; %s C" % temp)
```

[couleurs_melangees.py](#)

```
#PMIX_COLOUR

from sense_hat import SenseHat
sense = SenseHat()
r = 255
g = 255
b = 255
sense.clear ((r,g,b))
```

[eteindreleds.py](#)

```
#Eteindre les LED
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

[envahisseurs_espace.py](#)

```
#!/usr/bin/python
from sense_hat import SenseHat

sense = SenseHat()
sense.clear()
# importer l'image " space_invader.png" dans le repertoire du programme
/usr/src/sensehat/examples
sense.load_image("space_invader.png")
```

[arc_en_ciel.py](#)

```
#!/usr/bin/python
import time
from sense_hat import SenseHat

sense = SenseHat()
```

```
pixels = [  
    [255, 0, 0], [255, 0, 0], [255, 87, 0], [255, 196, 0], [205, 255,  
0], [95, 255, 0], [0, 255, 13], [0, 255, 122],  
    [255, 0, 0], [255, 96, 0], [255, 205, 0], [196, 255, 0], [87, 255,  
0], [0, 255, 22], [0, 255, 131], [0, 255, 240],  
    [255, 105, 0], [255, 214, 0], [187, 255, 0], [78, 255, 0], [0, 255,  
30], [0, 255, 140], [0, 255, 248], [0, 152, 255],  
    [255, 223, 0], [178, 255, 0], [70, 255, 0], [0, 255, 40], [0, 255,  
148], [0, 253, 255], [0, 144, 255], [0, 34, 255],  
    [170, 255, 0], [61, 255, 0], [0, 255, 48], [0, 255, 157], [0, 243,  
255], [0, 134, 255], [0, 26, 255], [83, 0, 255],  
    [52, 255, 0], [0, 255, 57], [0, 255, 166], [0, 235, 255], [0, 126,  
255], [0, 17, 255], [92, 0, 255], [201, 0, 255],  
    [0, 255, 66], [0, 255, 174], [0, 226, 255], [0, 117, 255], [0, 8,  
255], [100, 0, 255], [210, 0, 255], [255, 0, 192],  
    [0, 255, 183], [0, 217, 255], [0, 109, 255], [0, 0, 255], [110, 0,  
255], [218, 0, 255], [255, 0, 183], [255, 0, 74]  
]  
  
msleep = lambda x: time.sleep(x / 1000.0)  
  
def next_colour(pix):  
    r = pix[0]  
    g = pix[1]  
    b = pix[2]  
  
    if (r == 255 and g < 255 and b == 0):  
        g += 1  
  
    if (g == 255 and r > 0 and b == 0):  
        r -= 1  
  
    if (g == 255 and b < 255 and r == 0):  
        b += 1  
  
    if (b == 255 and g > 0 and r == 0):  
        g -= 1  
  
    if (b == 255 and r < 255 and g == 0):  
        r += 1  
  
    if (r == 255 and b > 0 and g == 0):  
        b -= 1  
  
    pix[0] = r  
    pix[1] = g  
    pix[2] = b  
  
while True:
```

```
for pix in pixels:
    next_colour(pix)

sense.set_pixels(pixels)
msleep(2)
```

boussole.py

```
#!/usr/bin/python
import sys
from sense_hat import SenseHat

# To get good results with the magnetometer you must first calibrate it
using
# the program in RTIMULib/Linux/RTIMULibCal
# The calibration program will produce the file RTIMULib.ini
# Copy it into the same folder as your Python code

led_loop = [4, 5, 6, 7, 15, 23, 31, 39, 47, 55, 63, 62, 61, 60, 59, 58,
57, 56, 48, 40, 32, 24, 16, 8, 0, 1, 2, 3]

sense = SenseHat()
sense.set_rotation(0)
sense.clear()

prev_x = 0
prev_y = 0

led_degree_ratio = len(led_loop) / 360.0

while True:
    dir = sense.get_compass()
    dir_inverted = 360 - dir # So LED appears to follow North
    led_index = int(led_degree_ratio * dir_inverted)
    offset = led_loop[led_index]

    y = offset // 8 # row
    x = offset % 8 # column

    if x != prev_x or y != prev_y:
        sense.set_pixel(prev_x, prev_y, 0, 0, 0)

    sense.set_pixel(x, y, 0, 0, 255)

    prev_x = x
    prev_y = y
```



Emulateur : un Sense-Hat sans Sense-Hat

Codes d'exemples en langage Python pour débiter:

[bonjourtoutlemonde.py](#)

```
#!/usr/bin/python
from sense_hat import SenseHat
sense = SenseHat()
sense.show_message("Hello world!")
```

[cycledecouleurs.py](#)

```
#!/usr/bin/python
import time
from sense_hat import SenseHat

sense = SenseHat()

r = 255
g = 0
b = 0

msleep = lambda x: time.sleep(x / 1000.0)

def next_colour():
    global r
    global g
    global b

    if (r == 255 and g < 255 and b == 0):
        g += 1

    if (g == 255 and r > 0 and b == 0):
        r -= 1

    if (g == 255 and b < 255 and r == 0):
        b += 1

    if (b == 255 and g > 0 and r == 0):
        g -= 1

    if (b == 255 and r < 255 and g == 0):
```

```

        r += 1

    if (r == 255 and b > 0 and g == 0):
        b -= 1

while True:
    sense.clear([r, g, b])
    msleep(2)
next_colour()

```

[defiletexte.py](#)

```

#!/usr/bin/python
from sense_hat import SenseHat

sense = SenseHat()
sense.set_rotation(180)
red = (255, 0, 0)
sense.show_message("One small step for Pi!", text_colour=red)

```

[laboussole.py](#)

```

#!/usr/bin/python
import sys
from sense_hat import SenseHat

# To get good results with the magnetometer you must first calibrate it
using
# the program in RTIMULib/Linux/RTIMULibCal
# The calibration program will produce the file RTIMULib.ini
# Copy it into the same folder as your Python code

led_loop = [4, 5, 6, 7, 15, 23, 31, 39, 47, 55, 63, 62, 61, 60, 59, 58,
57, 56, 48, 40, 32, 24, 16, 8, 0, 1, 2, 3]

sense = SenseHat()
sense.set_rotation(0)
sense.clear()

prev_x = 0
prev_y = 0

led_degree_ratio = len(led_loop) / 360.0

while True:
    dir = sense.get_compass()
    dir_inverted = 360 - dir # So LED appears to follow North
    led_index = int(led_degree_ratio * dir_inverted)
    offset = led_loop[led_index]

```



```
y = offset // 8 # row
x = offset % 8 # column

if x != prev_x or y != prev_y:
    sense.set_pixel(prev_x, prev_y, 0, 0, 0)

sense.set_pixel(x, y, 0, 0, 255)

prev_x = x
prev_y = y
```

From:

<http://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

http://chanterie37.fr/fablab37110/doku.php?id=sense_hat

Last update: **2023/01/27 16:07**

